#14 VT 8/30/04

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**In re Application of:**
**Timothy M. Young, Steven J. Capriotti, Steven Luzeski, Barbara E. Osder**

**Confirmation No.: 8021**

**Application No.: 09/363,339**

**Group Art Unit: 2465**

**Filing Date: July 29, 1999**

**Examiner: Simon P. Sing**

**For: VOICE MESSAGING SYSTEM WITH ENHANCED CUSTOMIZABILITY**

**EXPRESS MAIL LABEL NO: EV397435332US**
**DATE OF DEPOSIT: August 5, 2004**

Mail Stop Appeal-Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450



Sir:

**APPELLANT'S BRIEF PURSUANT TO 37 C.F.R. § 1.192**

This brief is being filed in support of Appellants' appeal from the rejections of Claims 1-25 in the Official Action dated January 5, 2004. Appellants filed a Notice of Appeal from the Examiner's final rejection on June 7, 2004. This appeal brief is being submitted in triplicate, pursuant to 37 C.F.R § 1.192(a).

Appellants respectfully request that the final rejection be reversed and that the application be remanded to the examining group for allowance.

## 1. REAL PARTY IN INTEREST

The real party in interest in the present appeal is Unisys Corporation by virtue of an assignment from the inventors (Timothy M. Young, Steven J. Capriotti, Steven Luzeski, and

Barbara E. Osder) (collectively the "Appellants") to Unisys Corporation which was filed on September 27, 1999 at Reel 010267, Frame 0592.

## 2. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to the Appellants, the Appellants' legal representative, or the Assignee that will directly affect or be directly affected by or have a bearing on the Board's decision in the present appeal.

## 3. STATUS OF CLAIMS

Claims 1 and 17 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Juster (U.S. Patent No. 5,724,406). Claims 1, 2, 8-15, 17, 18, and 20-24 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Sattar et al. (U.S. Patent No. 5,243,643). Claims 3-5 and 19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Matthews et al. (U.S. Patent No. 4,652,700). Claims 6 and 7 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Matthews and further in view of Weber (U.S. Patent No. 6,094,239). Claims 16 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Chencinski et al. (U.S. Patent No. 5,355,406).

## 4. PROSECUTION REFERENCES

The following references are deemed relevant to the analysis and arguments presented herein:

- the present application filed on July 29, 1999 (the "Application");
- the first Official Action dated October 24, 2002 (the "First Office Action");

- the Appellants' response to the First Office Action dated March 21, 2003 (the "First Response");
- the second Official Action dated June 19, 2003 (the "Second Office Action");
- the Appellants' response to the Second Office Action dated March 24, 2003 (the "Second Response");
- the third Official Action dated January 5, 2004 (the "Third Office Action");
- the Appellants' response to the Third Office Action dated April 5, 2004 (the "Third Response"); and
- the Advisory Action dated May 5, 2004 (the "Advisory Action").

## 5. STATUS OF AMENDMENTS

No amendments have been filed subsequent to the last rejection, and all previous amendments have been entered by the Examiner. Heretofore, Claims 1-2, 7, 10-11, 17-18, and 22-23, which were rejected in the First Office Action, were amended in the First Response. A complete list of the currently pending claims (as amended) are provided in Appendix A attached hereto.

## 6. SUMMARY OF INVENTION

Efficient and inexpensive customization of complex, generally large-scale telephony messaging systems benefit from the applicant's invention, but an understanding of such systems is required to understand the invention.

A telephony-based messaging system that provides voice, fax and/or e-mail messaging capabilities may comprise multiple servers each supporting a Network Applications Platform

( known as a NAP). A NAP comprises an underlying platform for storage and retrieval of messages and a messaging application running on the platform. A Voice Mail Application (VMA) is one example of a messaging application that may run on the NAP messaging platform. VMAs determine how calls to the messaging system are handled, what prompts are played to callers, and which features are available. In addition, a VMA typically maintains a database of subscribers who have mailboxes on the system. Received messages are stored by the VMA on the messaging platform in a local message store (or voice file). In operation, if a subscriber is not available when an incoming call is received, the call is forwarded to the NAP messaging system, which records a message from the caller under control of the VMA and then stores the message for later retrieval. A key, or token, returned to the VMA messaging application uniquely identifies the stored message data within the message store, and this key can be used at a later time to retrieve the message from the message store for playback to the subscriber. (Application, page 1, line 11 to page 2, line 16.)

The basic elements of a telephony messaging application or VMA include call flows, functions and prompts. Call flows define the logical flow of functions in the messaging application, and are often maintained in an interpretive state table. Functions (or code) are defined in these arts as the software building blocks used to implement the desired call flow functionality and, as such, comprise executable object code. Prompts are defined in these arts as logical groups or sequences of pre-recorded voice messages that often reside in a database and which are utilized to direct a caller through control elements of various call flows. (Application, page 7, lines 1-17.) The combination of a call flow and a function together to perform a specific task is referred to as a "module." (Application, page 14, lines 3-4).

Unfortunately, many messaging systems are inflexible and thus fail to meet the diverse needs and desires of the customer employing the messaging system, particularly in regard to the manner in which the messaging system interacts with callers. For example, if Customer A is a national telephone company in a country in which rotary phones are commonly used, Customer A may greatly prefer that the delimiter list (the list of DTMF tones or key presses a caller uses to signal the end of a caller input string) not include the pound (#) key since rotary phones do not include a pound key. Likewise, if Customer B is replacing an existing messaging system having call flow sequences that are familiar to its employees, Customer B may prefer to continue using these specific call flow sequences or functions instead of (or in addition to) whatever typical or default sequences and functions a messaging system may have. This may be particularly troublesome where corporate mergers occur and different systems need to be combined. Similarly, Customer B may also desire that specific DTMF tones/key presses correspond to certain functionality. Of course, such diverse customizations may be implemented by modifying the application code for each individual incarnation of the system; however, this solution (and others like it) is unsatisfactory because it creates complex customization and maintenance issues that are unduly expensive and tedious to implement, and likely to be inefficient in operation. (Application, page 4, lines 1-24).

The present invention, on the other hand, is directed to a messaging application (a VMA) that can be efficiently and inexpensively customized in accordance with the diverse requirements of customers. The invention provides for a modular, base version of a messaging application that is easily customized as needed on a module-by-module basis to meet the particular needs of individual customers. To accomplish this, a module of the present invention comprises not just call flows and function code but is unusual in that it also

includes at least one Customization List, thus enabling the module to be customized via modifications to its Customization List(s). A Customization List is used to tailor a set of call flows (and corresponding functions) to a particular purpose for a particular customer. (Application, page 4, line 26 to page 5, line 19 and page 17, lines 7-8.)

Each Customization List essentially comprises a table (or tables) with a list of names (corresponding to specific call flows and/or functions) and a modifiable list of corresponding DTMF signal identifiers (the telephone keys that a caller could press in response to a prompt), and a customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system (call flows and functions) by modifying the list of DTMF signal identifiers in the Customization List. Using these Customization Lists, the amount of call flow changes needed to customize an application to a particular customer is significantly reduced, and this approach reduces the field support and code maintenance involved with supporting multiple customizations of the base product. More specifically, the value of a DTMF digit is stored in the Customization List rather than hard-coded in a call flow. (Thus, DTMF signal key presses are stored outside of the call flow altogether). Therefore, when a customization requires different key presses to be used, the call flows do not have to change when using the applicant's invention, but, instead, only values in the Customization List need to be changed. (Application, page 14, lines 11-29; page 18, lines 2-7; and page 19, lines 16-18).

## 7. ISSUES

A. Whether Claims 1 and 17 are properly rejected under 35 U.S.C. § 102(b) as being anticipated by Juster or, more specifically, can a reference to enabling specific "call processing primitives" (CPPs) to caller-selected DTMF tones be found equivalent to allowing

for changable mapping between a customiztion list of modules and caller selected DTMF tones.

B. Whether Claims 1, 2, 8-15, 17, 18, and 20-24 are properly rejected under 35 U.S.C. § 102(b) as being anticipated by Sattar or, more specifically, whether a reference that does not have a customizable list separate and distinct from functions and/or call flows can be read on by claims that do.

C. Whether Claims 3-5 and 19 are properly rejected under 35 U.S.C. § 103(a) as being obvious and unpatentable over Sattar in view of Matthews.

D. Whether Claims 6 and 7 are properly rejected under 35 U.S.C. § 103(a) as being obvious and unpatentable over Sattar in view of Matthews further in view of Weber.

E. Whether Claims 16 and 25 are properly rejected under 35 U.S.C. § 103(a) as being obvious and unpatentable over Sattar in view of Chencinski.

## 8. GROUPING OF CLAIMS

Claims 1-25 stand or fall together.

## 9. SUMMARY OF PROSECUTION HISTORY

Independent Claim 1—which is also representative of the only other independent claim, Claim 17—originally claimed the following:

> 1. A telephony-based messaging system application stored on a computer readable medium for use by a particular customer, comprising:
>
> a module comprising call flows, code and a customization list;
>
> wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby the particular customer is

permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

In particular regard to the first limitation, and as set forth in the Specification, call flows define the logical flow of functions (or code) in the messaging application and are often maintained in an interpretive state table, whereas code (or "functions") are the software building blocks used to implement the desired call flow functionality and, as such, comprise executable object code (Application, page 7, lines 1-17).

In the First Office Action, the Examiner rejected the original Claim 1 under 35 U.S.C. § 112 for allegedly containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains to make and/or use the invention. Specifically, the Examiner, observing that "applicant claims a {software} module comprises or contains call flows," alleged that "[i]t is known in the art that a software program cannot comprise a call flow itself" and that "[a] software program can only comprise the functions of a call flow" (First Office Action, page 2, section 1.1).

To further prosecution, but without conceding the appropriateness of this rejection (particularly since call flows are often maintained in an interpretive state table and thus can comprise part of a software program), the Appellants amended Claim 1 as follows (with revisions shown):

1. A telephony-based messaging system application stored on a computer readable medium for use by a particular customer, comprising:

a module comprising call flow[s] <u>functions</u>, code and a customization list;

wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

This amendment to change "call flows" to "call flow functions" was specfically intended to clarify that what is claimed are call flows embodied in a software program such as in an interpretive state table described in the specification, and not just the idea or algorithm of call flows without being embodied in a tangible medium. Of course, **call flow functions are distinct and separate from code (a.k.a. functions), and both call flow functions and code are distinct and separate from customization lists**, and these distinctions are important to the Appellants' arguments presented herein below.

## 10. SUMMARY OF ARGUMENTS

Independent Claim 1—which is representative of all claims currently pending in the present Application—claims the following invention:

1. A telephony-based messaging system application stored on a computer readable medium for use by a particular customer, comprising:

a module comprising call flow functions, code and a customization list;

wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

For the invention described by the present claim, and as described in the specification of the present Application, DTMF values are stored in a distinct, separate, and modifiable customization list rather than hard-coded in a call flow or in the code—that is, DTMF signal key presses are stored outside of the call flows and code altogether. In this regard, Appellants respectfully submit that the novel utilization of a customization list patentably distinguishes the present invention from the prior art. The Examiner, however, has cited both Juster (U.S. Patent No. 5,724,406) and Sattar et al. (U.S. Patent No. 5,243,643) as allegedly each teaching prior art inventions that comprise a customization list.

In regard to Juster, the Examiner has heretofore alleged that "Juster's software {module} includes...a list of names (names of variables, functions, users, etc.) and a modifiable list of corresponding DTMF signal identifier" (First Office Action, page 3, section 4.1; *see also* Second Office Action, page 2, section 1.1; Third Office Action, page 2, section 1.1) and has cited the following as the basis for this allegation:

> For example, a user develops a voice mail messaging application having the necessary voice prompts and DTMF responses by selecting appropriate CPPs that generate those prompts and tone responses. ...
>
> Each CPP is implemented as a programming function, e.g. a "C" function using the C programming language, that can be executed by a suitable instruction execution means (i.e. the host CPU 14) and a data structure which serves as a descriptor for the CPP. However, this programming function is "invisible" to the user who simply configures a service using the primitives themselves. CPP descriptors are used by a state-table compilation and execution engine and include among other things (1) the CPP name, (2) information concerning the list of all the possible values the CPP can accept as

> arguments, and (3) the list of all possible events (results) the CPP can return when that CPP's execution is completed.

(Juster, col. 5, lines 23-26 and 41-53). More specifically (and most recently), the Examiner further stated that, in reference to Juster, the "Juster teaches a state table, which is a module for defining a state (DTMF) and a CPP to be executed (col. 9, lines 1-39)" (Advisory Action, page 2, section 1, paragraph 4). However, Appellants respectfully submit that the CPPs referred to in Juster are analogous to a mere combination of code (a.k.a., functions) and call flows without any kind of customization list that is both separate and distinguishable from code and call flows and not hard-coded directly into call flows and their corresponding functions in accordance with the prior art (see Application, page 19, lines 16-18). Specifically, Juster defines a CPP as follows:

> A call processing system in accordance with the present invention employs a call processing method for processing subscriber communications in a message handling system to provide a variety of different communications services. A set of **call processing primitives** is established to **perform various call processing services** in a messaging application, with **each primitive performing a single operation**. Sets of parameters are also defined for different types of call processing services and different subscribers.

(Juster, col. 2, lines 3-11) (emphasis added). Moreover, Juster's specific reference to a "state table" (or, more precisely, a "service state table") is in no way equivalent to a Customization List but, instead, to a call flow that invokes code. Indeed, Juster specifically states:

> The Service State Table is the principle component of the service definition. It represents the logical progression and execution of the call from beginning to end by means of states. Each state invokes one CPP and describes the transition to the next state based on the outcome of the CPP execution.

(Juster, col. 8., lines 8-13; *see also* col. 8, line 65 through col. 9, line 42). Therefore, Appellants respectfully submit that Juster at most teaches a system comprising calls flows and code, but nowhere does Juster teach the utilization of a Customization List as claimed in the present Application.

Likewise, in regard to Sattar, the Examiner has alleged that "Sattar discloses a voice processing system with configurable caller interfaces, comprising: a module {caller interface} comprising call flow functions, code and customization list (column 28, lines 18-37); wherein the customization list comprises a table of names and modifiable list of corresponding DTMF signal identifiers, where a customer is permitted to change the mapping between caller entered DTMF signal, and the corresponding actions taken by a voice messaging system (column 28, lines 1-3, 18-51; column 29-30, Vector Pogrecln)" (Second Office Action, page 3, section 1.1; *see also* Third Office Action, page 4, section 2.1). However, Appellants respectfully submit that, in the invention of Sattar, the DTMF signal mapping is still stored in the functions and/or call flows, for example, DTMF signal mapping stored in the software listing of Appendix A of Sattar which provides a listing of vectors (Sattar's equivalent to functions and/or call flows of the present application) used to control caller interfaces to the voicemail system (*see also* Sattar, col. 28, lines 6-10 and 18-24). Significantly, Sattar lacks any kind of customization list that is both separate and distinguishable from the vectors (code and call flows) and not hard-coded directly into the vectors in accordance with the prior art as claimed in the present Application (*see* Application, page 19, lines 16-18).

For these reasons, Appellants respectfully submit that the utilization of a customization list as claimed in the present Application is patentably distinguishable from the inventions disclosed in the Juster and Sattar references, and Appellents seek relief from the

Examiner's incorrect conclusions regarding the teachings of this prior art. A fully-developed analysis of the Examiner's position and Appellants arguments are presented in greater detail herein below.

## 11. DETAILED ANALYSIS AND PROSECUTION HISTORY

**Regarding Claims 1 and 17:**

Claims 1 and 17 were rejected under 35 U.S.C. § 102(b) as being anticipated by Juster (U.S. Patent No. 5,724,406).

In the Second Office Action, and in regard to Claim 1, the Examiner concluded that "Juster teaches using various call processing primitives (CPP) for customizing call process service (column 5, lines 12-32)" and that "Juster's software [module] includes functions of call flows (column 5, lines 26-29), codes (a software comprises computer executable codes), and a list of names (names of variables, functions, users, etc.) and a modifiable list of corresponding DTMF signal identifier (column 5, lines 23-26)." The Examiner also reaches a similar conclusion in regard to Claim 17.

In the Second Response, and without conceding the appropriateness of equating the "CPPs" of Juster to the "modules" of the invention in the present invention, Appellants respectfully submitted that Juster does not teach or suggest "a table with a list of names and a modifiable list of corresponding DTMF signal identifiers" as set forth in Claims 1 and 17 and referred to in the Application as a "Customization List" which is described as comprising "a table with a list of names recognized by the development system and a modifiable list of corresponding DTMF signals identifiers" that enable a "customer" to "change the mapping between caller-entered DTMF signal and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers in the Customization List(s)"

(Application, page 14, lines 24-29). In contrast, the invention of Juster comprises a plurality of CPPs (which, utlimately, are more equivalent to "functions" than to "modules" as these terms are used in the present Application) where each CPP "perform[s] one single identifiable operation, such as recording a message, playing a prompt, collecting a digit, reading DTMF sequences, etc." (Juster, col. 5, lines 27-29). Moreover, according to Juster, "a user develops a voice mail messaging application having the necessary voice prompts and DTMF responses *by selecting appropriate CPPs [modules] that generate those prompts and tone responses*" (col. 5, lines 23-26) (emphasis added). In other words, each CPP in the invention of Juster has hard-coded DTMF signal identifiers, and thus it would be necessary to select an alternate CPP having the same functionality but different DTMF signals when a user desires to change which DTMF signals initiate the desired CPP functionality. Therefore, to avoid modifying CPPs (like the invention of the present application avoids the need to modify the functions/"code"), the invention of Juster would necessarily require a library of CPPs, one for each possible combination and/or permutation of possible DTMF signal identifiers that a customer may find desirable to use to call the specific functionality. Yet nowhere does Juster suggest that a user should modify a CPP such that specific DTMF signals corresponding to specific functionality for that specific CPP are thereby changed, but instead teaches away from a DTMF-modifiable CPP by implying that a user must select a specific (or "appropriate") CPP to achieve specific functionality, including hard-coded DTMF mapping, which is predefined for each individual CPP. Consequently, the inability of a user to change the specific DTMF signals corresponding to specific functionality for a specific CPP in the invention of Juster requires a library of CPPs for each possible combination of DTMF signals mapped to specific CPP functionality. The only other alternative is custom development of CPPs on an as-needed basis where "such diverse requirements are handled by modifying the

application code as required for each different incarnation...[t]his solution, however, is unsatisfactory since it creates a complex maintenance dilemma...[t]he changes may be simple in nature but the maintenance of a set of changes to the standard product for every customer can nonetheless become unduly burdensome and expensive" (Application, page 4, lines 13-21), and thus the invention of Juster fits the description of the insufficient prior art as described by the present Application.

Returning to the present Application, Claim 1 recites "a module comprising call flows, code and a Customization List; wherein the Customization List comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby *the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers*" (emphasis added). For example, the customer might have a preferred numbers combination pertaining to specific functionality (for example, pressing "*6" to delete a voicemail message), or that the conversion from letters to numbers on the telephone keypad could provide for mnemonic enhancement for the user's benefit (for example, pressing "335"—the numbers corresponding to the letters "DEL"—to delete a voicemail message). The benefits afforded by the present invention are simply not realizable using nothing more than the teachings of Juster—namely that a Customization List enables a user to quickly modify the specific DTMF signals corresponding to specific functionality for that specific module, and thereby preclude the need for a library of modules with various DTMF signal mappings. This element patentably distinguishes the present invention from the invention of Juster, and Claim 17, which includes limitations similar to those of Claim 1, is likewise patentably distinguishable from the teachings of Juster for the same reasons.

To further clarify the differences between Juster and the invention of the current Application, Appellants respectfully submit a point-by-point analysis of the Examiner's key comments in the "Response to Arguments" section of the Third Office Action in which the Examiner maintains his rejection of Claims 1 and 17. For convenience, the Examiners comments from the Third Office Action are italicized, and the Appellants' arguments drawn from the Third Response, hereafter follow:

- *"The applicant argues that Juster does not disclose a table linking a DTMF identifier and its corresponding action, and DTMF identifiers are hard-coded into CPPs (functions or modules), thus requires a library of CPPs, one of each possible combination/permutation of possible DTMF signal identifiers with various functions, and making Juster's system lacking flexibility."* (Third Office Action, page 9, lines 15-19.)

  On the contrary, the Appellants do not broadly contend that Juster's system lacks flexibility in the absolute, but only does so relative to the invention of the present Application. More accurately, it is the Appellants' contention that the system of Juster does indeed provide *some* flexibility, but only to the extent that multiple CPPs of identical functionality for different DTMF responses are available and provided to a customer.

  For example, if the Juster invention has a CPP for Functionality A mapped to DTMF "1", another CPP for Functionality A mapped to DTMF "2", and a third CPP for Functionality A mapped to DTMF "3", then the user in Juster would have three choices for which DTMF signal would be used to call Function A. However, if the system of Juster does not have yet another CPP for Functionality A mapped to DTMF "4", then the user could not select—and therefore would not be able to use—DTMF "4" to call Function A because the DTMF signal for a specific CPP is hard-coded in that CPP and

cannot be changed by a person who would utilize the invention (i.e., a customer).

In contrast, Appellants' claimed invention enables a customer to map (and re-map) specific DTMF signal identifiers to specific modules (each of which performs specific desired functions) where remapping DTMF signals to the modules requires nothing more than modifying a Customization List table. Thus, in the present Application, Function A needs only be embodied in a single module, but that single module can then be associated with any specific DTMF signals the customer desires. A library of CPPs under the Juster teachings would need to be very large to approach the same level of flexibility. Simply put, Juster lacks any teaching or suggestion of a "customization list [that] comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers," as required by Appellants' Claim 1.

- *"However, Juster teaches: 'a user develops a voice mail messaging application having the necessary voice prompts and DTMF responses by selecting appropriated [sic] CPPs that generate those prompts and tone responses' (column 5, lines 23-26) (also quoted by the applicant in the Remark, page 3)."* (Third Office Action, page 9, lines 20 to page 10, line 2.)

Appellants have quoted this section of the Juster reference both for what it says and what it does not say. More specifically, a user must select an appropriate CPP to generate the prompts and tone responses—not "change" or "modify" or "remap" the DTMF signals for a single CPP. Selecting an appropriate CPP necessarily means that there are a plurality from which to choose, and thus a selection inherently implies that no modifications to any CPP are occurring when "a user develops a voice mail messaging

application" (id.). Therefore, the Examiner's reliance on this section to sustain the position that Juster teaches a means for modifying CPPs is unfounded.

- *"Juster clearly teaches that the voice mail messaging system outputs voice prompts and responses (action taken by the messaging system) in responds [sic] to a DTMF identifier entered by a user, and by inherency, it is a table linking DTMF identifiers to their corresponding actions (defined by CPPs) as Juster states: 'each CPP performs one simple identifiable operation, such as recording a message, playing a prompt, collecting a digit, reading DTMF sequence etc' (column 5, lines 27-29)."* (Third Office Action, page 10, lines 2-8.)

While Juster may in fact teach a voice mail messaging system that outputs voice prompts and responses in response to a DTMF identifier entered by a user, the Examiner's position that, "by inherency", there must be a table linking DTMF identifiers to their actions is erroneous. As previously explained, the user of Juster's system must select the CPPs with the desired functionality and having the desired DTMF signals, which he forms into a linked list. If the user (of Juster) later wants to use a different DTMF signal to call specific functionality, the user must replace the existing CPP in the linked list with another—e.g., replace CPP-A1 (Functionality A, DTMF "1") with CPP-A5 (Functionality A, DTMF "5") in the system.

Clearly, the table of Juster is not a "customization list [that] comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers," as required by Appellants' Claim 1..

- *"Juster clearly teaches a friendly environment in that a user, without programming knowledge, may customize a link list (a table, see page 8, lines 17-24 of*

*applicant's Application) of DTMF identifiers and corresponding actions by selecting a desired CPP (action) corresponds [sic] to a particular DTMF identifier (column 4, lines 49-54; column 5, lines 23-26, 45-53)."* (Third Office Action, page 10, lines 8-12.)

Appellants once again point out that "changes" to the function-based table in Juster require the removal of old CPPs and the addition of new CPPs in the linked list, whereas in the invention of the present Application the modules remain the same but only the DTMF data in the Customization List is changed.

In addition to the foregoing, and to even further clarify the patentably-distinct differences between Juster and the invention of the current Application, Appellants also respectfully submit a point-by-point analysis of the Examiner's key comments in the "Response to Arguments" section of the Advisory Action in which the Examiner maintains his rejection of Claims 1 and 17. For convenience, the Examiners comments from the Advisory Action are italicized, and the Appellants' arguments to same presented for this Appeal thereafter follow:

- *"Examiner believes that the claimed table is the 'result table' described on page 18, lines 20-24 of the Application. The Application explains that the mapping between result path names and the digits pressed (DTMF in this case) is the result table."* (Advisory Action, page 2, lines 6-8).

  The "table with a list of names and a modifiable list of corresponding DTMF signal identifiers" of Claim 1 is indeed a "result table" as that term is defined in the Application. However, Appellants respectfully submit that what is claimed is the utilization of a "Customization List" that comprises a result table, and that this Customization List is ***distinct and separate*** from the call flows and the code—the three components that

together constitute a "module" in the present invention—as illustrated in Fig. 4 and as described in the present Application as follows: "Each Service Module includes call flows 42-1, code 42-2, one or more Standard Customization Lists 42-3, and one or more Customized Customization Lists 42-4. Each Customization List includes tables or lists of result names and digits, break names and digits, delimiter names and digits, and double-digit names and digits. In other words, each Customization List comprises a table with a list of names recognized by the development system and a modifiable list of corresponding DTMF signal identifiers (i.e., the telephone keys that a caller could press.) As discussed, a particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers in the Customization List(s)." (Application, page 14, lines 18-29.) In addition, the Application also states the following: "When a customization requires a different key presses to be used, the call flows do not have to change. Only the result table needs to be changed." (Application, page 18, lines 22-24.) **Therefore, the "state table" of the present invention, which comprises the DTMF signal identifier mapping, is part of the "Customization List" which, in turn, is distinct and separate from the "call flow" and "code" for the invention of the present Application.**

- *"Juster teaches a state table, which is a module for defining a state (DTMF) and a CPP to be executed (column 9, lines 1-39). ..."* (Advisory Action, page 2, lines 11-12.)

While Juster uses the term "state table" (or, more precisely, the term "service state table"), Appellants respectfully disagree with the Examiner in that this term, as used in Juster, is in no way equivalent to what is described by the use of this the term "state table"

in the present Application. Specifically, Juster states the following: "The Service State Table [elsewhere, simply the "service state table"] is the principle component of the service definition. It represents the logical progression and execution of the call from beginning to end by means of states. Each state invokes one CPP and describes the transition to the next state based on the outcome of the CPP execution." (Juster, col. 8., lines 8-13; see also col. 8, line 65 through col. 9, line 42.)

Significantly, this is almost the exact definition of a "call flow" as used in the present Application, which states the following: "Call flows define the personality of the application, i.e., the logical flow of the application interface to the caller (the person that dialed a phone number and was transferred to the telephony application in question). Functions represent the software building blocks employed to implement the desired call flow functionality. A function is computer code and may involve access to an application database. Each function produces a result that determines the path of the associated call flow." (Application, page 7, lines 3-9.) Based on these two passages, **a person of skill in the art would conclude that the term "state table" as used in Juster describes what is termed a "call flow" in the present Application.** Furthermore, based on these same passages, **a person of skill in the art would also conclude that the term "CPP" as used in Juster is analogous to the term "function" (or "code") as this term is used in the present Application.** However, Juster nowhere teaches the third necessary element of a Customization List or its equivalent as claimed in the present Application.

- *"Furthermore, the state table is part of the Subscriber Specified Parameters, which can vary from customer to customer."* (Advisory Action, page 2, lines 12-13.)

Without conceding the validity of this conclusion, Appellants respectfully submit that,

if indeed true, this variance from customer to customer is due to changes in the what Juster terms a "state table" (that is, what the present Application terms the "call flows") as represented by the pseudocode of Juster, col. 9, lines 1-24. However, the present invention is directed to a system that enables a customer to change the DTMF signal identifiers without having to change the call flows by instead revision of a Customization List. "When a customization requires different key presses to be used, the call flows do not have to change. Only the result table need to be changed." (Application, page 18, lines 22-24.) Thus, "[u]sing these Customization Lists, the amount of call flow changes needed to customize an application to a particular customer is significantly reduced" (Application, page 14, lines 13-15), where the amount of the reduction is equivalent to the changes that would otherwise necessarily have to be made to the call flow to effect DTMF remapping.

- *"Juster also teaches that a user may define the state table, and may modify call services having the necessary voice prompts and DTMF responses by selecting appropriate CPPs that generate those prompts and tone responses (column 5, lines 12-29). Therefore, a customer can modify a state table to change DTMF entered to CPPs executed."* (Advisory Action, page 2, lines 13-17.)

Without conceding the validity of this conclusion, Appellants respectfully submit that, if indeed true, the selection of CPPs—which, again, are the equivalent to "functions" (or "code") in the present application—by the state table of Juster—which, again, is the equivalent of a "call flow" in the present application—does not in any way utilize a Customization List, as that term is defined by the present Application, as Juster does not disclose anything equivalent to the Customization List of the present Application.

In summary, Appellants respectfully submit that the invention of Juster at most comprises call flows ("server state tables") and functions/code ("CPPs"), but nowhere does Juster teach, as stated in Claim 1, the utilization of a "Customization List comprising a table with a list of name and a modifiable list of corresponding DTMF signal identifies" which, by definition, must exist distinctly and separately from the CPPs and the server state tables as illustrated in Figure 4 of the present application. By existing separately from the pseudocode of the call flows (the server state tables), a customer is able to modify the DTMF mapping without having to changing the call flows.

Moreover, Appellants respectfully submit that only by impermissibly viewing the Appellants' disclosure in hindsight could a person of skill in the art read into the Juster reference the functionality and benefits described and disclosed in the present Application, and that absent the present Application the Juster reference in no way teaches a system where DTMF signals are mapped to modules in a Customization List.

For the foregoing reasons, Appellants respectfully submit that the inventions recited in Claims 1 and 17 fully comply with the requirements of 35 U.S.C. § 102. Appellants therefore request that this patent application be remanded to the Examiner with an instruction to both withdraw the rejections for alleged unpatentability of Claims 1 and 17 and allow said claims.

**Regarding Claims 1, 2, 8-15, 17, 18, and 20-24:**

Claims 1, 2, 8-15, 17, 18, and 20-24 were rejected under 35 U.S.C. § 102(b) as being anticipated by Sattar et al. (U.S. Patent No. 5,243,643).

In the Second Office Action, and in regard to independent Claim 1 (upon which Claims 2 and 8-15 directly or indirectly depend), the Examiner concludes that "Sattar discloses a voice processing system with configurable caller interfaces, comprising: a module

[caller interface] comprising call flow functions, code and customization list (column 28, lines 18-37); wherein the customization list comprises a table of names and modifiable list of corresponding DTMF signal identifiers, where a customer is permitted to change the mapping between caller entered DTMF signal, and the corresponding actions taken by a voice messaging system (column 28, lines 1-3, 18-51; column 29-30, Vector Pogrecln)." The Examiner also reaches a similar conclusion in regard to Claim 17.

In the Second Response, Appellants respectfully submitted that Juster does not teach or suggest "a table with a list of names and a modifiable list of corresponding DTMF signal identifiers" as set forth in Claims 1 and 17 and referred to in the Application as a "Customization List" which is described as comprising "a table with a list of names recognized by the development system and a modifiable list of corresponding DTMF signals identifiers" that enable a "customer" to "change the mapping between caller-entered DTMF signal and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers in the Customization List(s)" (Application, page 14, lines 24-29). Thus, in the invention of the present application, the DTMF signal identifiers are mapped to specific modules, each of which performs specific desired functions, and remapping DTMF signals to the modules requires merely modifying the Customization List. The invention of Sattar, in contrast, takes an entirely different approach where the DTMF signal mapping is stored in functions and/or call flows, for example, DTMF signal mapping stored in the software listing of Appendix A of Sattar (see also Sattar, col. 28, lines 18-24). Appendix A is (a) a listing of vectors (Sattar's equivalent to functions and/or call flows of the present application) used to control caller interfaces to the voicemail system (see col. 28, lines 6-10). These vectors are stored in an Application State Logic Table (AST) which, unlike a Customization List, is compiled and used in object code form (see col. 11, line 66 through

col. 12, line 10) and that are generated by an application editor (APE) (see col. 14, lines 67 through col. 15, line 19). Significantly, vectors are only modifiable and adaptable in the C-language (col. 11, lines 42-49; col. 12, lines 21-26), and thus changing vectors requires recompilation. More specifically, to change DTMF signals, a "user"—which is actually an application developer (see col. 28, lines 1-6)—edits the vectors using APE (col. 28, lines 16-24) which, again, is an editor for source code that must be compiled for use.

Nowhere does Sattar suggest modifying a single vector such that specific DTMF signals corresponding to specific functionality for that specific vector are thereby changed (aside from implying an ability to rewrite the C-language code and recompiling). On the contrary, Sattar instead teaches away from a DTMF-modifiable vector and speaks only to an application developer hard-coding DTMF mapping in the C-programming language for compilation (thus resulting in a predefined vector from the end-user perspective). Consequently, the inability of a user to change the specific DTMF signals corresponding to specific functionality for a specific vector in the invention of Sattar requires a programmer to first develop, for utilization by an end-user, a library of vectors for each possible combination of DTMF signals mapped to specific vector functionality from which, presumably, the user can select the DTMF mapping desired. The only other alternative is custom development of vectors on an as-needed basis where "such diverse requirements are handled by modifying the application code as required for each different incarnation...[t]his solution, however, is unsatisfactory since it creates a complex maintenance dilemma...[t]he changes may be simple in nature but the maintenance of a set of changes to the standard product for every customer can nonetheless become unduly burdensome and expensive" (Application, page 4, lines 13-21).

In contrast, Claim 1 of the present invention claims "a module comprising call flows, code and a customization list; wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby *the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers*" (emphasis added). To recite an example previously used herein, a customer might have preferred numbers pertaining to specific functionality (for example, pressing "*6" to delete a voicemail message), or that the conversion from letters to numbers on the telephone keypad could provide for mnemonic enhancement for the user's benefit (for example, pressing "335"—the numbers corresponding to the letters "DEL"—to delete a voicemail message). The benefits afforded by the present invention are not available via the invention of Sattar.

This specific limitation directed to the utilization of a Customization List, which enables a user to modify the specific DTMF signals corresponding to specific functionality for that specific module, precludes the need for other alternatives such as a library of vectors with various DTMF signal mapping (per the alternate but dissimilar solution suggested for Juster earlier herein), and thereby clearly differentiates the present invention from the invention of Sattar. Moreover, Claim 17 includes a similar limitation, and is likewise distinguishable from the teachings of Sattar.

Furthermore, to draw parallels from the analysis of the Juster reference, nowhere does Sattar teach that a customization list that is distinct and separate from the call flows and functions/code, whereby the DTMF signal mapping can be modified without having to

change the call flows. This is an expressed feature and advantage of the invention of the present Application which is both explicitly and implicitly absent in the invention of Sattar.

Lastly, in regard to the point-by-point analysis of the Third Office Action and the Advisory Action presented in regard to the Juster reference earlier herein, Appellants respectfully submit that similar arguments and analysis apply with equal force to the Sattar reference though, for economy, said analysis and arguments will not be repeated herein.

In light of the reasons discussed herein above as well as those pertaining to the Juster reference which apply to the Sattar reference with equal force, Appellants respectfully submit that only by impermissibly viewing the Appellants' disclosure in hindsight could a person of skill in the art read into the Sattar reference the functionality and benefits described and disclosed in the present Application, and that absent the present Application the Sattar reference fails to teach a system where DTMF signals are mapped to modules in a Customization List, a base component element of all claims at issue here.

For the foregoing reasons, Appellants respectfully submit that the inventions recited in Claims 1-2, 8-15, 17-18, and 20-24 fully comply with the requirements of 35 U.S.C. § 102. Appellants therefore request that this patent application be remanded to the Examiner with an instruction to both withdraw the rejections for alleged unpatentability of Claims 1-2, 8-15, 17-18, and 20-24 and allow said claims.

**Regarding Claims 3-7, 16, 19, and 25:**

Claims 3-7, 16, 19, and 25 have been rejected as follows: Claims 3-5 and 19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Matthews et al. (U.S. Patent No. 4,652,700); Claims 6 and 7 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Matthews and further in view of Weber (U.S. Patent

No. 6,094,239); and Claims 16 and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sattar in view of Chencinski et al. (U.S. Patent No. 5,355,406). However, the shortcomings of Sattar are not overcome by the teachings of Matthews, Weber, and/or Chencinski, alone or in combination, as none of these references are directed to a system whereby DTMF mapping can be accomplished by a customer lacking programming skills by modifying a customization list that is distinct and separate from the calls flows and the functions/code. Moreover, Claims 3-7, 16, 19, and 25 depend from independent Claim 1 or Claim 17 and, as such, are allowable as dependent claims depending from an allowable claim.

For the foregoing reasons, Appellants respectfully submit that the inventions recited in Claims 3-7, 16, 19, and 25 fully comply with the requirements of 35 U.S.C. § 103. Appellants therefore requests that this patent application be remanded to the Examiner with an instruction to both withdraw the rejections for alleged unpatentability of Claims 3-7, 16, 19, and 25 and allow said claims.

## 12. SUMMARY OF ARGUMENTS AND CONLCUSIONS

As described in detail in the previous section, independent Claim 1—which is representative of all claims currently pending in the present Application—claims a telephony-based messaging system application comprising a module comprising call flow functions, code and a customization list wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, and whereby the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

For the invention described by the present claim, and as described in the specification of the present Application, DTMF values are stored in a distinct, separate, and modifiable customization list rather than hard-coded in a call flow or in the code—that is, DTMF signal key presses are stored outside of the call flows and code altogether. In this regard, Appellants respectfully submit that the novel utilization of a customization list patentably distinguishes the present invention from the prior art. The Examiner, however, has cited both Juster (U.S. Patent No. 5,724,406) and Sattar et al. (U.S. Patent No. 5,243,643) as allegedly each teaching prior art inventions that comprise a customization list.

In regard to Juster, the Examiner has heretofore alleged that "Juster's software {module} includes...a list of names (names of variables, functions, users, etc.) and a modifiable list of corresponding DTMF signal identifier" (First Office Action, page 3, section 4.1; *see also* Second Office Action, page 2, section 1.1; Third Office Action, page 2, section 1.1), and that "Juster teaches a state table, which is a module for defining a state (DTMF) and a CPP to be executed (col. 9, lines 1-39)" (Advisory Action, page 2, section 1, paragraph 4). However, Appellants respectfully submit that the CPPs referred to in Juster are analogous to a mere combination of code (a.k.a., functions) and call flows without any kind of customization list that is both separate and distinguishable from code and call flows and not hard-coded directly into call flows and their corresponding functions in accordance with the prior art (see Application, page 19, lines 16-18). Moreover, Juster's specific reference to a "state table" (or, more precisely, a "service state table") is in no way equivalent to a Customization List but, instead, to a call flow that invokes code as Juster specifically states that the "Service State Table is the principle component of the service definition" that "represents the logical progression and execution of the call from beginning to end by means of states" and that "[e]ach state invokes one CPP and describes the transition to the next state based on the

outcome of the CPP execution." (Juster, col. 8., lines 8-13; see also col. 8, line 65 through col. 9, line 42). Therefore, Appellants respectfully submit that Juster at most teaches a system comprising calls flows and code, but nowhere does Juster teach the utilization of a Customization List as claimed in the present Application.

Likewise, in regard to Sattar, the Examiner has alleged that "Sattar discloses a voice processing system with configurable caller interfaces, comprising: a module {caller interface} comprising call flow functions, code and customization list..." (Second Office Action, page 3, section 1.1; *see also* Third Office Action, page 4, section 2.1). However, Appellants respectfully submit that, in the invention of Sattar, the DTMF signal mapping is still stored in the functions and/or call flows, for example, DTMF signal mapping stored in the software listing of Appendix A of Sattar which provides a listing of vectors (Sattar's equivalent to functions and/or call flows of the present application) used to control caller interfaces to the voicemail system (*see also* Sattar, col. 28, lines 6-10 and 18-24)—in other words, Sattar lacks any kind of customization list that is both separate and distinguishable from the vectors (code and call flows) and not hard-coded directly into the vectors in accordance with the prior art as claimed in the present Application (*see* Application, page 19, lines 16-18).

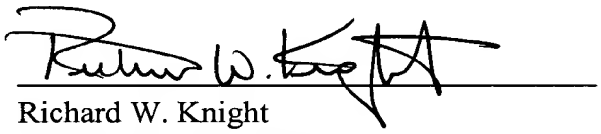
***[Remainder of Page Intentionally Left Blank]***

## CONCLUSION

Based on the detailed arguments and analysis presented above, Appellants respectfully submit that the utilization of a customization list as claimed in the present Application is patentably distinguishable from the inventions disclosed in the Juster and Sattar references, and Appellents seek relief from the Examiner's incorrect conclusions regarding the teachings of this prior art. More specifically, Appellants submit that the inventions recited in claims 1-25 fully comply with the requirements of 35 U.S.C. § 102 and § 103, and Appellants therefore request that this patent application be remanded to the Examiner with an instruction to both withdraw the rejections for alleged unpatentability and allow the appealed claims.

Respectfully submitted,

Date: August 5, 2004



Richard W. Knight
Registration No. 42,751

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Correspondence Address:
Unisys Corporation
Attn: Office of the General Counsel (Patents)
Unisys Way, MS/E8-114
Blue Bell, PA 19424-0001

## APPENDIX A

**Claims On Appeal:**

1. A telephony-based messaging system application stored on a computer readable medium for use by a particular customer, comprising:

a module comprising call flow functions, code and a customization list;

wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

2. A telephony-based messaging system application as recited in claim 1, and further comprising a main module comprising code and call flow functions that are specific to the particular customer.

3. A telephony-based messaging system application as recited in claim 1, wherein the customization list comprises a set of tables, with one table for each of the following: result list names, break list names, delimiter list names and double-digit list names.

4. A telephony-based messaging system application as recited in claim 3, wherein a result list name refers to a component of a call state that determines the next call state in a call flow.

5. A telephony-based messaging system application as recited in claim 3, wherein a break list name refers to a component of a call state that effects a stoppage of a prompt.

6. A telephony-based messaging system application as recited in claim 3, wherein a delimiter list name refers to a component of a call state that determines the end of a caller input string.

7. A telephony-based messaging system application as recited in claim 3, wherein a double-digit is a sequence of two DTMF tones that a caller enters within a predetermined period, and wherein a double-digit list name refers to a menu in which the corresponding double-digit sequence is recognized as a single result.

8. A telephony-based messaging system application as recited in claim 1, wherein the module is susceptible to being independently built into a set of flat files that can be loaded during an application initialization process.

9. A telephony-based messaging system application as recited in claim 1, wherein the customization list includes a standard customization table and a customized customization table derived from the standard table.

10. A telephony-based messaging system application as recited in claim 1, wherein the module is a main module that contains call flow functions and code that are specific to the particular customer.

11. A telephony-based messaging system application as recited in claim 1, wherein the module is a service module that contains call flow functions and code that are used by multiple customers.

12. A telephony-based messaging system application as recited in claim 1, wherein the storage medium is a computer readable magnetic storage medium.

13. A telephony-based messaging system application as recited in claim 1, wherein the storage medium is a computer readable optical storage medium.

14. A telephony-based messaging system application as recited in claim 1, wherein the storage medium is a computer network.

15. A telephony-based messaging system application as recited in claim 1, wherein the application is a voice and/or fax mail application.

16. A telephony-based messaging system application as recited in claim 1, wherein the application is a bank by phone application.

17. A telephony-based messaging system, comprising:

a computer;

a network interface unit coupling the computer to a telephone network;

a network applications platform (NAP) running on the computer; and

a messaging application comprising a module containing call flow functions, code and a customization list, wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby a customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers.

18. A telephony-based messaging system as recited in claim 17, wherein the messaging application further comprises a main module comprising code and call flow functions that are specific to the particular customer.

19. A telephony-based messaging system as recited in claim 17, wherein the customization list comprises a set of tables, with one table for each of the following: result list names, break list names, delimiter list names and double-digit list names.

20. A telephony-based messaging system as recited in claim 17, wherein the module is susceptible to being independently compiled into a set of flat files that can be loaded during an application initialization process.

21. A telephony-based messaging system as recited in claim 17, wherein the customization list includes a standard customization table and a customized customization table derived from the standard table.

22. A telephony-based messaging system as recited in claim 17, wherein the module is a main module that contains call flow functions and code that are specific to the particular customer.

23. A telephony-based messaging system as recited in claim 17, wherein the module is a service module that contains call flow functions and code that are used by multiple customers.

24. A telephony-based messaging system as recited in claim 17, wherein the application is a voice and/or fax mail application.

25. A telephony-based messaging system as recited in claim 17, wherein the application is a bank by phone application.